(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 992 909 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
12.04.2000 Bulletin 2000/15

(51) Int Cl.[7]: **G06F 11/14**

(21) Application number: **99307381.6**

(22) Date of filing: **17.09.1999**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **30.09.1998 US 163708**

(71) Applicant: **NCR INTERNATIONAL INC.**
Dayton, Ohio 45479 (US)

(72) Inventors:
- **Robinson, Irving M.**
  San Diego CA 92127 (US)
- **Antoun, Selim Zoher**
  San Diego CA 92126 (US)
- **Dempster, Pataric B.**
  Atlantic Highlands NJ 07716 (US)
- **MacDonald, Robert J.**
  San Diego CA 92129 (US)
- **Stellwagen Jr., Richard G.**
  Poway CA 92064 (US)
- **Blevins, Terence J.**
  Springboro OH 45066 (US)
- **Ramsey, David Allen**
  Lexington SC 29072 (US)

(74) Representative: **Cleary, Fidelma et al**
International IP Department
NCR Limited
206 Marylebone Road
London NW1 6LY (GB)

(54) Provision of continuous database service and scalable query performance using active redundant copies

(57) A method and apparatus for distributing computer resources in a network environment. A network of computer systems is partitioned into at least one computing system partition, and is configured into at least one redundancy group. The computing system partitions include applications, computing system nodes, and copies of a database schema. The copies of the database schema are replicated at each computing system partition within a network. The computing system partition manages interactions between the instances, the computing system nodes, and the copy of the database schema within the respective computing system partition. The redundancy group comprises at least one computing system and at a plurality of computing system partitions, and manages the replication of the database schema within the computing system and computing system partitions.

EP 0 992 909 A2

Description

[0001] The invention relates in general to computer systems, and more particularly, to a provision for continuous database service using active redundant copies of the database.

[0002] Many modern computer systems are in nearly continuous use, and have very little time to be taken "down" or "offline" for database updates or preventative maintenance. Further, computer systems increasingly require systems that virtually never fail and have little or no scheduled downtime. As a concurrent requirement, these same systems demand cost-effective computing solutions, open systems to avoid or reduce specific supplier dependencies, and the ability to leverage the latest hardware and software technologies as they become available.

[0003] Modern computer systems also have transitioned from a static installation to a dynamic system that regularly changes. The system continually contains new collections of products that are processing requests from a constantly changing user base. The ability of computing solutions to provide service availability in a dynamic environment is becoming increasingly important, because the pace of change in products and customers' environments is expected to increase. The term "change tolerance" has been used to describe the ability of a computer system to adapt to the dynamic environment required.

[0004] It can be seen, then, that there is a need in the art for a system that provides a high confidence level for continuous processing. It can also be seen, then, that there is a need in the art for a system with a high change tolerance. It can also be seen, then, that there is a need in the art for a system with reasonable development costs and implementation schedules that does not sacrifice the benefits of open systems.

[0005] It is the object of the present invention to provide a system which overcomes the limitations in the prior art as described above;

[0006] According to a first aspect, the present invention discloses a system for providing continuous database access, comprising:

at least one computing system partition, including at least one instance of an application, at least one computing system node, and at least one copy of a database schema, the copies of the database schema being replicated at each computing system partition within a network, and wherein each computing system partition manages interactions between the instances, the computing system nodes, and the copy of the database schema within the respective computing system partition;

a plurality of computing systems connected together via the network, wherein each computing system comprises one or more computing system partitions;

at least one redundancy group, comprising at least one computing system and a plurality of computing system partitions, wherein each redundancy group manages the replication of the database schema within the computing system and computing system partitions within the redundancy group.

[0007] The computing systems are connected together via the network, and comprise one or more computing system partitions. The redundancy group comprises at least one computing system and at a plurality of computing system partitions, and manages the replication of the database schema within the computing system and computing system partitions.

[0008] According to a second aspect, this present invention resides in a method for distributing computer resources in a network environment, comprising the steps of:

assembling, as part of a computer network, at least one computing system partition, including at least one instance of an application, at least one computing system node, and at least one copy of a database schema, the copies of the database schema being replicated at each computing system partition within the computer network;

configuring, within the computer network, a plurality of computing systems connected together via the computer network, wherein each computing system comprises one or more computing system partitions;

configuring, within the computer network, at least one redundancy group, comprising at least one computing system and at a plurality of computing system partitions; and

managing at least one task using the database schema within the computing system and computing system partitions with the redundancy group.

[0009] According to a third aspect, the present invention resides in a method for providing database access, comprising the steps of:

operating at least one computing system within a network, the computing system containing at least one computing system partition and the computing system being a member of a redundancy group, wherein the computing system partition includes at least one instance of an application, at least one computing system node, and at least one copy of a database schema, the copies of the database schema being replicated at each computing system partition within a network; and

managing the replication of the database schema within the computing system and computing system partitions within the redundancy group.

[0010] An embodiment of the present invention will now be described by way of reference to the accompanying drawings in which:

[0011] Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIG. 1 is a block diagram that illustrates an exemplary hardware environment that could be used with the present invention;

FIG. 2 illustrates the computing system partition of the present invention;

FIG. 3 illustrates the redundancy strategy of the present invention;

FIG. 4 illustrates a model of the computer architecture of the present invention;

FIG. 5 illustrates replication of the database using the present invention;

FIG. 6 illustrates temporal consistency of the database that is propagated by the present invention; and

FIG. 7 is a flowchart that illustrates exemplary logic performed by the controller according to the present invention.

Overview

[0012] The present invention discloses a method, apparatus, and article of manufacture for distributing computer resources in a network environment.

[0013] At least one redundancy group and at least one computing system partition within the redundancy group are defined. The redundancy group comprises selected computer resources within the network environment. The selected computer resources include at least one application, at least one computer node, and at least one copy of a database. The computing system partitions comprise a subset of the selected computer resources within the network environment. The subset of the selected computer resources including at least one application, at least one computing node, and using only one copy of the database within the network environment. Tasks are performed within the network using the first computing system partition to perform the task.

[0014] Control for the detection of reassigning the recipient of the updates can occur upon hardware or software problems with the first assignee, or to allow the first assignee to be taken out of service for maintenance purposes. This control is provided by a combination of software systems operating on each of the networked computer systems, and can also be provided on external computers called Control Computers. The software on the networked computer and control computer together determine the status of each of the networked computers to determine when to reassign the recipient computer, and if so, which of the networked computers should receive the database updates. The determination is achieved by using periodic messages, time-out values, and retry counts between the software on the networked computers and the control computers.

Hardware Environment

[0015] FIG. 1 is an exemplary hardware environment used to implement the preferred embodiment of the invention. The present invention is typically implemented using a plurality of computing systems 100A-100D, each of which generally includes, inter alia, a processor, random access memory (RAM), data storage devices (e.g., hard, floppy, and/or CD-ROM disk drives, etc.), data communications devices (e.g., modems, network interfaces, etc.), monitor (e.g., CRT, LCD display, holographic display device, etc.), etc.

[0016] The computing systems 100A-100D are coupled together via network 102 and comprise a redundancy group 104. Each computing system 100A-D further comprises one or more computing system partitions, which are described in further detail in FIGS. 24. In addition, management centers 106A and 106B can be coupled to network 102. Management centers 106A and 106B are representative only; there can be a greater or lesser number of management centers 106 in the network 102. Further, there can be a greater or lesser number of computing systems 100A-100D connected to the network 102, as well as a greater or lesser number of computing systems 100A-D within the redundancy group 104.

[0017] The present invention also teaches that any combination of the above components, or any number of different components, including computer programs, peripherals, and other devices, may be used to implement the present invention, so long as similar functions are performed thereby. The presentation of the computer system as described in FIG. 1 is not meant to limit the scope of the present invention, but to illustrate one possible embodiment of the present invention.

Relationships and Operation

[0018] FIG. 2 further illustrates the components within a computing system of the present invention. Within the computing systems 100A-D are one or more computing system partitions (CSPs) 202. Each CSP 202 is coupled to only one copy of a database 204. The computing systems 100A-D are coupled together via network 102.

[0019] Management center computer 106A (or, alternatively, 106B) can be used to control the flow of data from the database copies 204 and updates to the computer systems 100A-100D. The database 204 can also be controlled directly from computer systems 100A-D if desired.

[0020] Each copy of the database 204 is associated with a computing system partition (CSP) 202. As shown in FIG. 2, each computer system 100A-D can have one or more CSP's 202 resident within a computer system, as illustrated in computer system 100A.

[0021] A redundancy. group 104 is a collection of Computing System Partitions 202 (CSPs) collaborating in an actively redundant fashion on a specific workload using a single replicated database 204 schema. The CSPs 202 may be resident on a single node computer system 100, a multi-node computer system 100, or on selected subsets of computing nodes from one or more multi-node computer systems 100. Each CSP 202 has an independent database copy of the database 204 for the redundancy group 104. The definition for a CSP 202 is that set of computing resources using a single copy of the replicated database 204.

[0022] The fundamental component of a CSP 202 is a single computing node executing an independent copy of an operating system. However, CSP 202 may consist of multiple nodes and, therefore, multiple operating system instances. The operating system operating on each CSP 202 can be different, e.g., one CSP 202 may be using Windows, while another CSP 202 uses Unix, etc. An operating system instance may be a participant in one and only one redundancy group 104, meaning that the computing nodes comprising a CSP 202 are "owned" by that redundancy group 104. A multi-node system can have different nodes participating in different redundancy groups 104, but there must be no overlap between redundancy groups 104.

[0023] To synchronize and replicate the database 204 between the computer systems 100A-100D, one ofthe computer systems 100A-D is responsible for receiving direct updates of the database 204 via network 102 and disseminating or replicating those updates of database 204 to the remaining computing systems 100.

[0024] As an example, computer system 100B can be designated as the recipient ofthe direct updates to database 204. Once the updates are received by computer system 100B, computer system 100B then sends a copy ofthe database 204 with updates to computer systems 100A, 100C, and 100D via network 102. This process continues until computer system 100B has sent a copy of database with updates to all computer systems within the network 102.

[0025] If computer system 100B is unavailable, the responsibility of replicating the database and updates shifts to another computer in the network 102. As an example, if computer system 100B is unavailable, the database replication responsibility shifts to computer system 100C, which then receives direct updates. Computer system 100C then replicates the database and updates to computer systems 100A and 100D. Computer system 100C continues the replication until all computers in the network 102 receive copies of the database and updates.

Redundancy Strategy

[0026] FIG. 3 illustrates the hierarchical redundancy strategy of the present invention. To effectively perform the replication of the database 204 and the updates as described in FIG. 2, the present invention partitions the network 102 into redundancy groups 104. Each redundancy group 104 is comprised of computing systems 100, computing system partitions 202, application instances 302, computing system nodes 304, and database copy 306. Database copy 306 is comprised of database 204 and updates.

[0027] Typical networks 102 have multiple redundancy groups 104. The relationship between redundancy groups 104 is somewhat limited, but all redundancy groups 104 can participate in a global network 102, and a global administration view is typically used for such a network 102. In general, however, redundancy groups 104 are envisioned to be mostly independent of each other and constructed for the purposes of application-level independence, administrative flexibility, or the ability to use computing systems 100 of modest capabilities.

[0028] The redundancy group 104 is the fundamental factor of service availability and scalable query performance. The present invention uses the redundancy group 104 to reduce or eliminate a server service outage so long as at least one CSP 202 in the redundancy group 104 is fully operational. The present invention also uses the redundancy group 104 to scale query performance beyond that attainable with just one computing system partition 104 and one copy of the database 306. Query performance and availability scale as CSP's 202 are added to a redundancy group. With standard computers, as performance goes up, availability typically goes down. The present invention allows both availability and query performance for data warehousing systems to both go up simultaneously.

[0029] Redundancy groups 104 of the present invention accommodate the condition in which CSPs 202 arbitrarily undergo exit and reintroduction scenarios, but a sufficiently configured redundancy group 104 does not cease proper functionality. The limits of redundancy group functionality and database 204 access is limited by scenarios outside of the control of the computer system, e.g., unplanned hardware or software malfunctions, etc.

Computer Architecture Model

[0030] FIG. 4 illustrates a model of the computer architecture of a computing system partition of the present invention. The architecture model 400 has three significant environments: the management environment 402, the run-time environment 404, and the hardware environment 406. The management environment 402 is illustrated as redundancy group management. The run-time environment 404 comprises the software components that provide application services directly or indirectly, which is the majority of the components in the model 400. The hardware environment 406 is depicted as the hardware platform, e.g., computer network 102, and peripherals.

[0031] Redundancy group management 402 com-

prises of the tools, utilities and services necessary to administer, supervise and provide executive control over elements of a redundancy group 104. The components within the redundancy group management 402 environment include redundancy group administration 408, redundancy group supervision 410, redundancy group execution 412.

[0032] The redundancy group administration 408 component provides tools for definition, configuration, and operations of a redundancy group 104. These tools communicate with other tools that provide administrative control of product specific components. Operations include facilities to startup, shutdown, install, and/or upgrade elements of redundancy groups 104. Included in the upgrade and install categories are special facilities necessary for verification. Included in the definition and configuration capabilities are defining policies and procedures to be used by both humans and machines. Additionally, it is foreseen that advanced utilities to determine the scope of failures and subsequently identify recovery procedures would be in this component.

The tools of this environment are expected to execute in a management center facility.

[0033] The redundancy group supervision 410 component provides those services that monitor the health of a redundancy group 104. Included are the services for status request handling, heartbeat setup and monitoring, and failure detection.

[0034] The redundancy group execution 412 component provides those executive services that manage and control the work load of a redundancy group. Included are those services that provide transaction and request-level load balancing and reconfiguration. This component manages and controls the workload of normal transactions as well as recovery requests.

Run-time Environment

[0035] The run-time environment 404 comprises the services necessary to support application programs within redundancy groups 104. The components of the run-time environment 404 include application execution services 414, applications 416, communications resource services 418, global transaction services 420, shared resource services 422, database replication services 424, file i/o 426, remote storage services 428, and network services 430. These components fall into two categories, 1) those components typically utilized by applications directly, and 2) those components typically utilized by applications indirectly. Services that fall into the second category are used by those services in the first category.

[0036] Application execution services 414 provide pre- and post-processing on behalf of an application 416. Such services include application instantiation, parameter marshaling, and queue access services. Application execution services 414 also inform the application 416 of the status of a given transaction request and its disposition; for example, whether it is a normal transaction request, a recovery request, or whether the request is a request to startup or shutdown the application. Application execution services 414 also include services necessary to communicate to redundancy group management 402 components. Additionally, application execution services 414 handle application 416 error situations.

[0037] Applications 416 are services to the consumers of a system (network 102), and are composed of software components. Applications 416 are reduced in complexity by leveraging other services in a rich operating environment, such as application 416 execution services 414 and shared resource services 422, since these other services supply needed levels of transparency.

[0038] The communication resource services 418 component comprises services that provide application 416-to-application 416 communications within redundancy groups.

[0039] The global transaction services 420 component provides services to maintain transaction context and to coordinate transaction integrity procedures and protocols. These services include facilities for an application 416 to query the global transaction status, and commit or abort transactions.

[0040] The shared resource services 422 component is a general container for services that provide access to shared resources. In a redundancy group 104 the shared resources of interest are replicated databases 204, and, therefore, database 204 access services reside in the shared resource services 422 component. Database 204 access services include services that provide the capability to create, read, write, rewrite, and delete data within a replicated database.

[0041] Database replication services 424 fall into the indirect class of application 416 services. The database replication services 424 propagate database updates transparently to all copies of the database 204 in a redundancy group 104. There are primarily two database 204 replication models, as described in the discussion relating to FIG. 5.

[0042] File i/o services 426 are not utilized directly by customer applications 416, but are provided for use by system software components requiring non-transactional, persistent data storage and access services. File i/o is typically used for logging or journaling functions, event capture, software executables, and data interchange files.

[0043] Remote storage services 428 allow a given file update request to be processed at locations remote from the location ofthe file i/o request, enabling file replication. System components that take advantage of these services are those that require non-transactional access to queues, logs and system files that would be inappropriate for storage in an database.

[0044] Network services 430 include those services that provide high performance, highly reliable transport

of messages. Of specific interest are those services that provide multi-casting of messages which results in an optimal and guaranteed delivery of messages to all destinations in a specified domain of receivers. This component also benefits applications indirectly, e.g., customer applications would not call the interface that initiates these services. Rather, these services would be provided to the application 416 through communications resource services 418.

[0045] Network platform 406 is the computing hardware, e.g., network 102, that is used for executing the instructions associated with the application 416, etc.

Database Replication Schemes

[0046] FIG. 5 illustrates replication of the database using the present invention. Within network 424, replication schemes 500 and 502 can be utilized to replicate database 204. Either replication scheme 500 or replication scheme 502, or both, can be used within network 424, depending on the architecture of the redundancy groups 104.

[0047] Database 204 replication is the synchronization mechanism between the database 204 copies in a redundancy group 104. The present invention could also utilize transaction-level replication (reprocessing the entire application transaction on each participating system) instead of entire database 204 replication, but the discussion relating to database 204 replication applies equally well to transaction-level replication. References herein relating to database 204 replication include transaction-level replication.

[0048] At least two distinct database 204 replication models are supported by the present invention, peer/peer replication model 500 and primary/subscriber replication model 502. Other database replication models are envisioned, but the discussion herein is limited to the two models 500 and 502. The peer/peer replication model 502 update transactions are processed on any logical system in a redundancy group 104. Inter-copy database 204 consistency and serializability are maintained either through global network 102 concurrency controls 504, or through commit certifications that occur within the redundancy group 104.

[0049] In the primary/subscriber replication model 502, all update transactions are routed to a single logical system, e.g., computer system 100A-D, in the redundancy group 104, called the primary system, which propagates updates to the other logical systems, e.g., computer systems 100A-D, after the commitment of a transaction is complete. The update transaction routing is performed transparently and automatically. When the primary logical system, e.g., computer system 100A, exits the redundancy group 104 (for reasons of failure or scheduled downtime) a new primary system is selected. See the discussion relating to FIG. 2.

[0050] FIG. 6 illustrates temporal consistency of the database that is propagated by the present invention. Within either replication model 500 or 502, the database 204 will have temporal inconsistencies because time is required to update the database 204 on each of the network 102 computers within a redundancy group 104. Update propagation in replicated database 204 processing has a side effect in that a trade-off must be made between update efficiency and the temporal consistency of thedatabase 204 copies in the redundancy group 104. It is possible to synchronize the database 204 copies by propagating updates before the completion of an update transaction, e.g., before releasing database 204 locks and allowing commit processing to complete. However, absolute synchronization requires propagation protocols that are complex and expensive from a computing perspective.

[0051] The present invention allows the database 204 copies to deviate from each other in a temporal sense, and restrict consistency constraints to serializability and transaction-level atomicity. The approach of the present invention prevents any copy of the database 204 from having "dirty data," "partial updates," or out-of-order updates, but the timing ofthe appearance of the updates from a given transaction in any particular database 204 copy will be delayed to an unpredictable degree. The temporal deviation between the database 204 copies will be dependent on numerous factors including hardware utilization, instantaneous transaction mix, and network 102 latency.

The effects of inter-copy temporal inconsistency can be mitigated with numerous application processing techniques, including restriction of updates to selected time windows (during which queries may be restricted), clever partitioning of the query processing workload, and clever partitioning and/or clustering of user queries to specific database copies.

[0052] For a single replicated database schema, shown in replication model 502, each actively redundant configuration will support only one replicated database schema because oftransaction-level consistency constraints.

Logic of the Database Replicator

[0053] FIG. 7 is a flowchart that illustrates exemplary logic performed by the controller according to the present invention.

[0054] Block 700 represents the computer 100 assembling, as part of a computer network, at least one computing system partition, including at least one instance of an application, at least one computing system node, and at least one copy of a database schema, the copies of the database schema being replicated at each computing system partition within the computer network.

[0055] Block 702 represents the computer 100 configuring, within the computer network, a plurality of computing systems connected together via the computer network, wherein each computing system comprises

one or more computing system partitions.

[0056] Block 704 represents the computer 100 configuring, within the computer network, at least one redundancy group, comprising at least one computing system and at a plurality of computing system partitions.

[0057] Block 706 represents the computer 100 managing at least one task using the database schema within the computing system and computing system partitions with the redundancy group.

Conclusion

[0058] This concludes the description of the preferred embodiment of the invention. The following describes some alternative embodiments for accomplishing the present invention. For example, any type of computer, such as a mainframe, minicomputer, or personal computer, could be used with the present invention. In addition, any software program utilizing (either partially or entirely) a database could benefit from the present invention.

[0059] The foregoing description of the preferred embodiment of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto.

Claims

1. A system for providing continuous database access, comprising:

   at least one computing system partition, including at least one instance of an application, at least one computing system node, and at least one copy of a database schema, the copies of the database schema being replicated at each computing system partition within a network, and wherein each computing system partition manages interactions between the instances, the computing system nodes, and the copy of the database schema within the respective computing system partition;

   a plurality of computing systems connected together via the network, wherein each computing system comprises one or more computing system partitions;

   at least one redundancy group, comprising at least one computing system and a plurality of computing system partitions, wherein each redundancy group manages the replication of the database schema within the computing system and computing system partitions within the redundancy group.

2. The system of claim 1, wherein the redundancy group defines a first computing system as the computing system that replicates the copy of the database to all other computing systems within the redundancy group.

3. The system of claim 2, wherein the redundancy group defines a second computing system within the redundancy group for replicating the copy of the database to all other computing systems within the redundancy group when the first computing system is unavailable.

4. The system of claim 1, wherein the redundancy group can be redefined to include a different set of computing systems.

5. The system of claim 1, wherein the computing system partition can be redefined to include different instances of the application and different computing system nodes.

6. The system of claim I, wherein at least one computing system partition can be removed from the redundancy group.

7. The system of claim 6, wherein the computing system partitions that are removed from the redundancy group can be added to a second redundancy group.

8. The system of claim 1, wherein a status of the computing system partitions is monitored by the redundancy group.

9. The system of claim 8, wherein a task that is assigned to a first computing system partition is reassigned to a second computing system partition by the redundancy group.

10. A method for distributing computer resources in a network environment, comprising the steps of:

    assembling, as part of a computer network, at least one computing system partition, including at least one instance of an application, at least one computing system node, and at least one copy of a database schema, the copies of the database schema being replicated at each computing system partition within the computer network;

    configuring, within the computer network, a plurality of computing systems connected together via the computer network, wherein each computing system comprises one or more computing system partitions;

    configuring, within the computer network, at least one redundancy group, comprising at

least one computing system and at a plurality of computing system partitions; and
managing at least one task using the database schema within the computing system and computing system partitions with the redundancy group.

11. The method of claim 10, wherein the task is a database replication within the computer network.

12. The method of claim 11, wherein the task of database replication is performed by a first computing system partition within the redundancy group.

13. The method of claim 12, wherein the task of database replication is performed by a second computing system partition within the redundancy group when the first computing system partition is unavailable.

14. The method of claim 10, wherein the redundancy group can be redefined to include a different set of computing systems.

15. The method of claim 10, wherein the computing system partition can be redefined to include a different subset of computing system nodes and instances of the application.

16. The method of claim 10, wherein a first computing system partition can be removed from the redundancy group.

17. The method of claim 16, wherein the first computing system partition can be added to a second redundancy group.

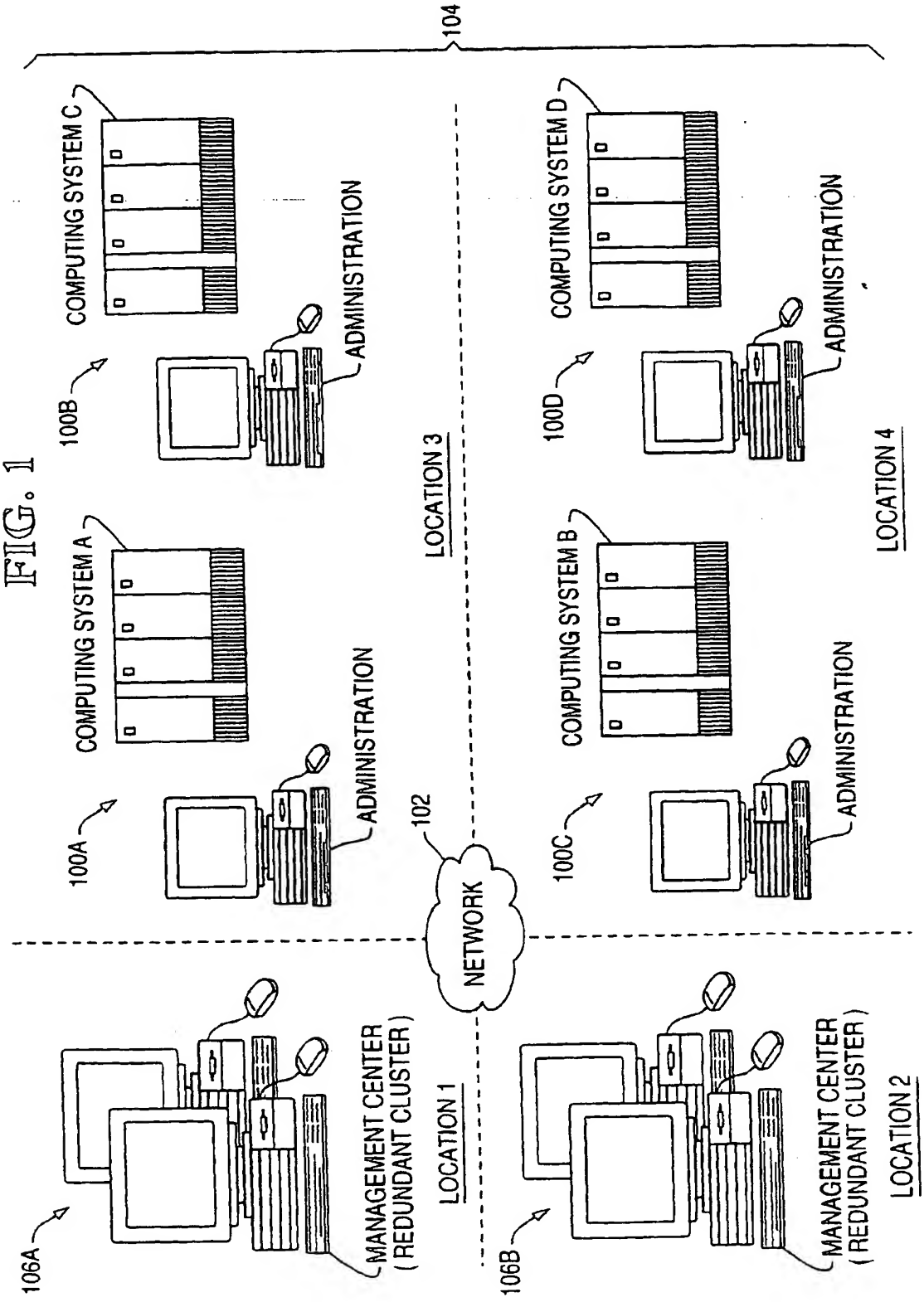
18. The method of claim 10, further comprising the step of monitoring a status of the computing system partitions by the redundancy group.

19. The method of claim 18, wherein a task that is assigned to a first computing system partition is reassigned to a second computing system partition by the redundancy group.

20. A method for providing database access, comprising the steps of:

operating at least one computing system within a network, the computing system containing at least one computing system partition and the computing system being a member of a redundancy group, wherein the computing system partition includes at least one instance of an application, at least one computing system node, and at least one copy of a database schema, the copies of the database schema being replicated at each computing system partition within a network; and
managing the replication of the database schema within the computing system and computing system partitions within the redundancy group.

FIG. 1



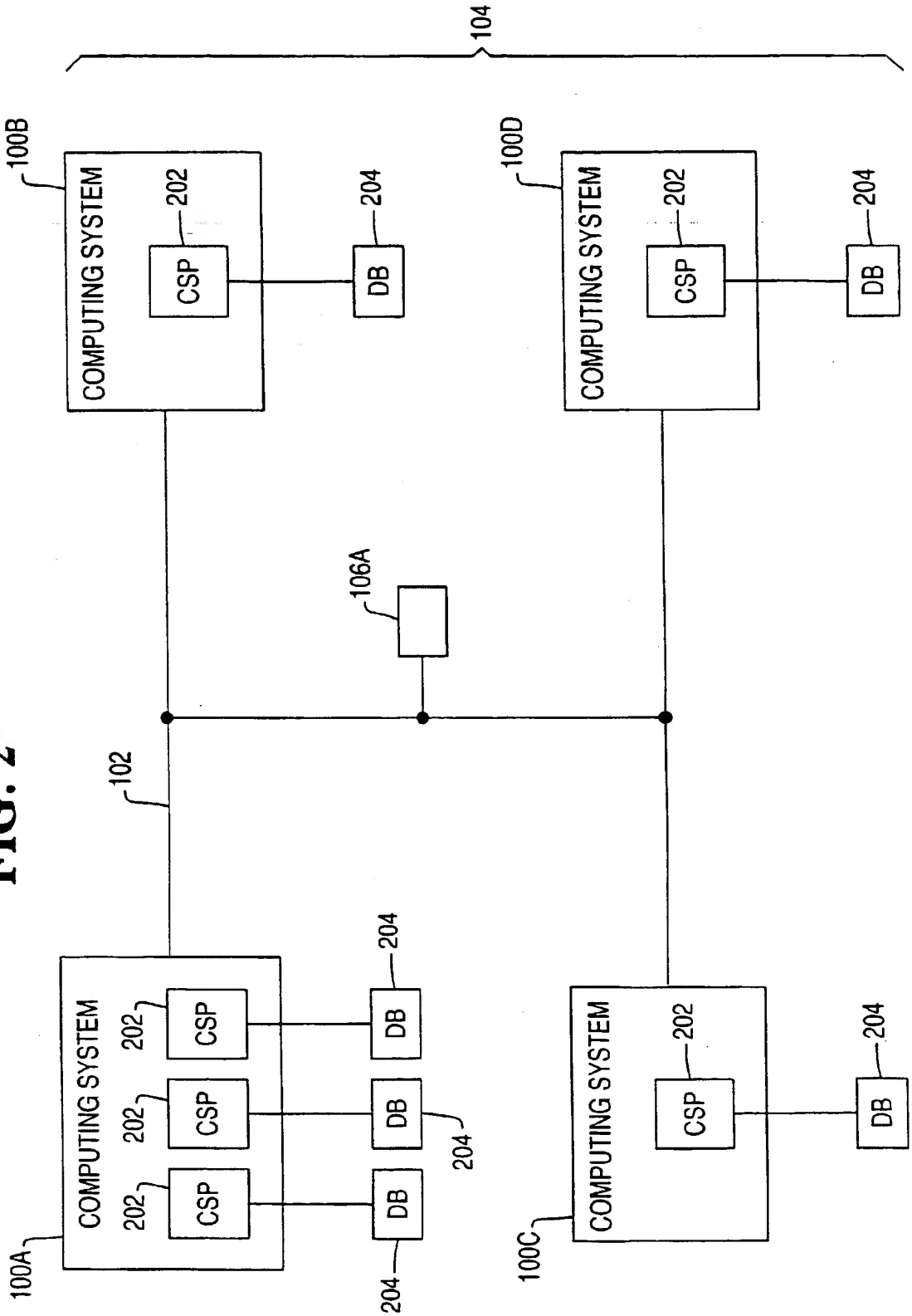


FIG. 2

FIG. 3

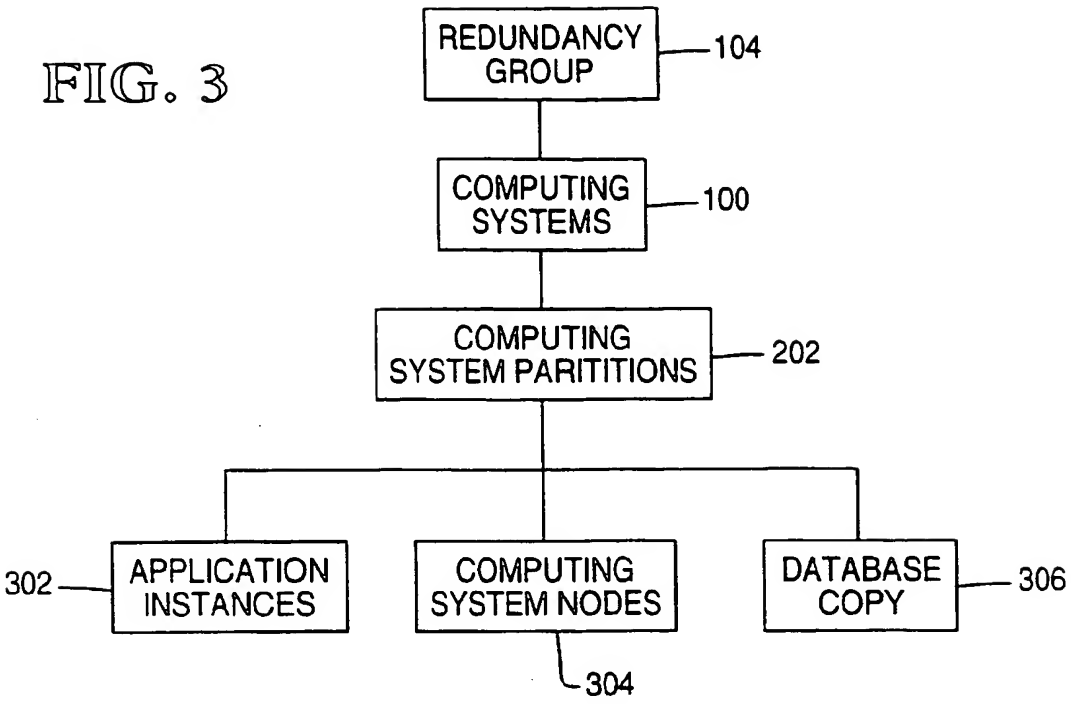


FIG. 7

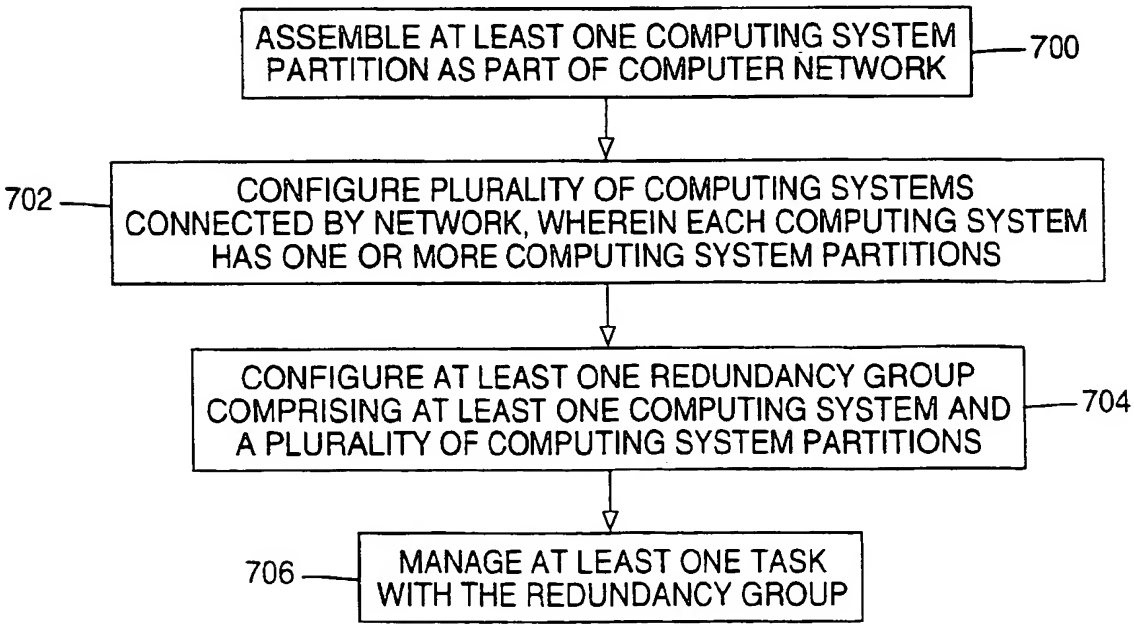


FIG. 4

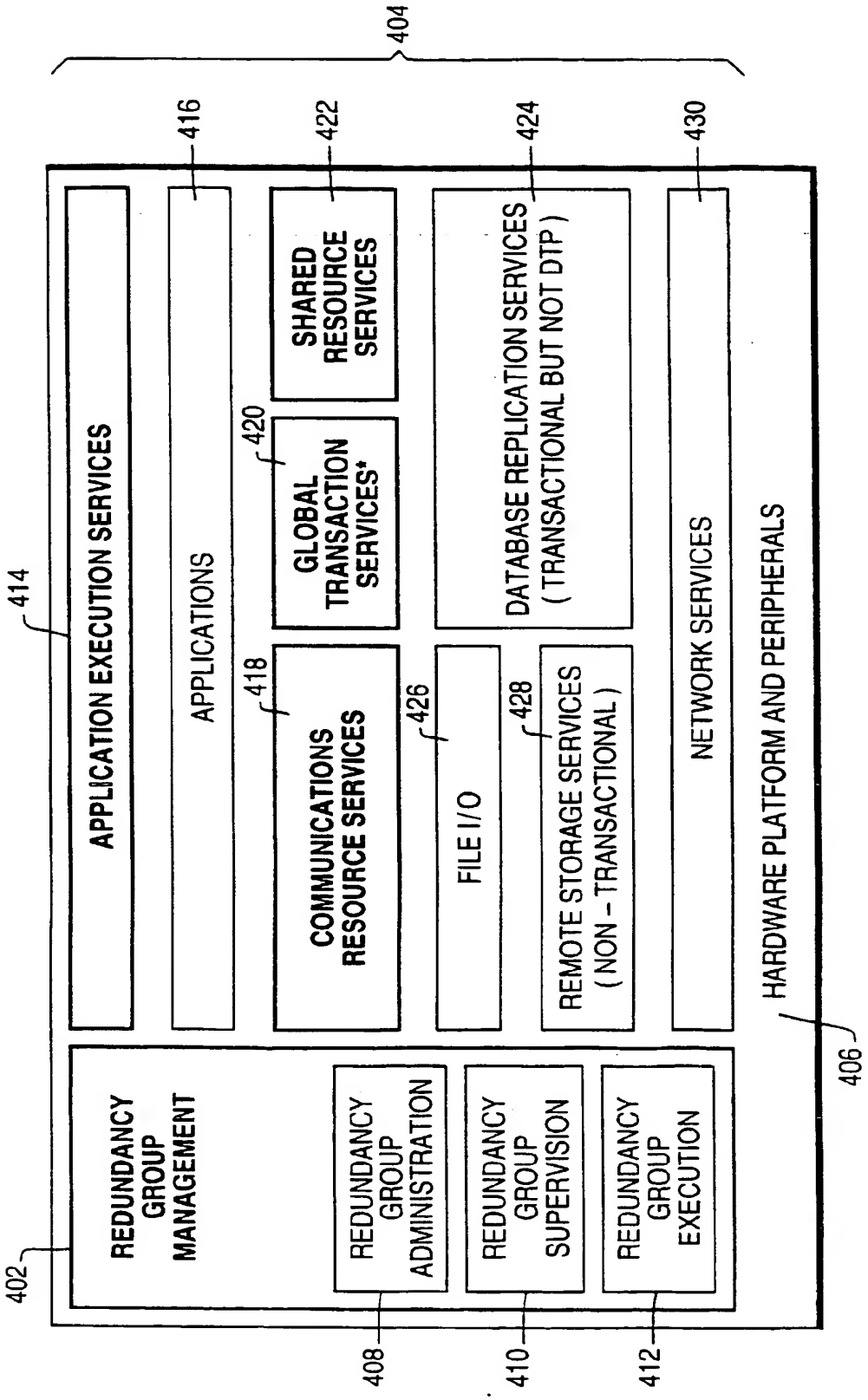


FIG. 5

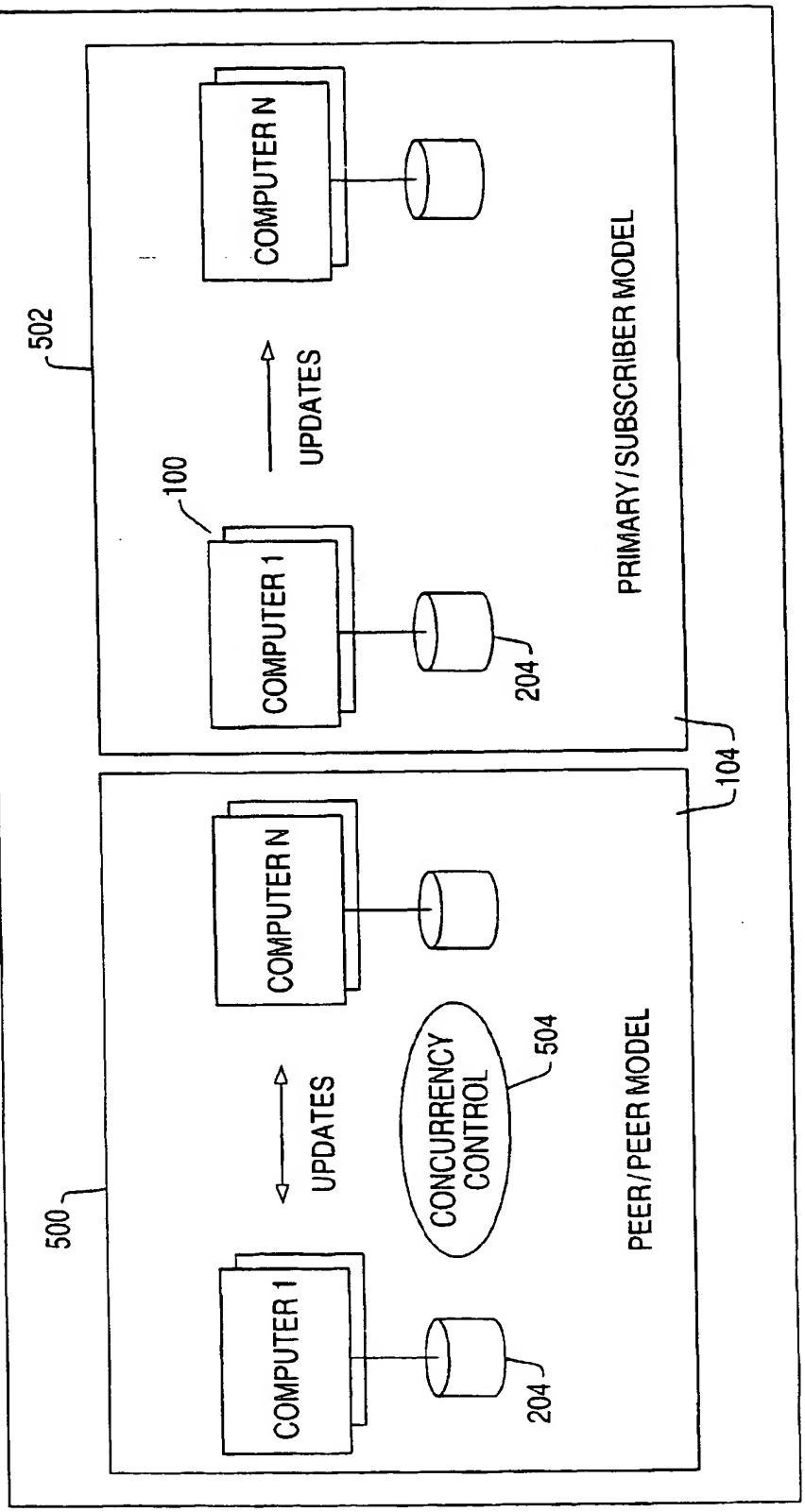
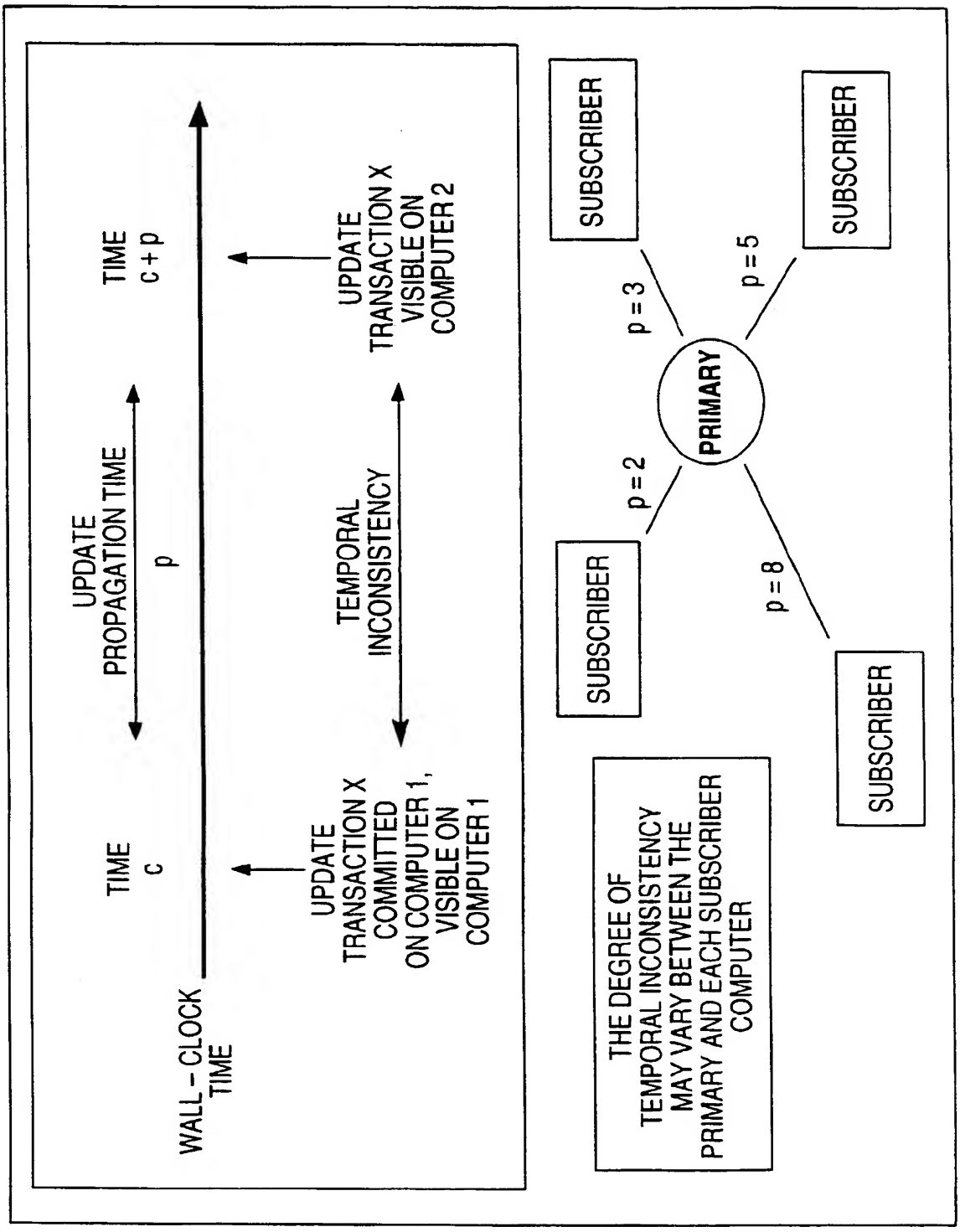


FIG. 6



(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 992 909 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
19.12.2001 Bulletin 2001/51

(43) Date of publication A2:
12.04.2000 Bulletin 2000/15

(21) Application number: **99307381.6**

(22) Date of filing: **17.09.1999**

(51) Int Cl.[7]: **G06F 11/14**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **30.09.1998 US 163708**

(71) Applicant: **NCR INTERNATIONAL INC.**
Dayton, Ohio 45479 (US)

(72) Inventors:
- **Robinson, Irving M.**
  San Diego CA 92127 (US)
- **ANTOUN, Selim Zoher**
  Del Mar, CA 92014 (US)
- **Dempster, Pataric B.**
  Atlantic Highlands NJ 07716 (US)
- **MacDonald, Robert J.**
  San Diego CA 92129 (US)
- **Stellwagen Jr., Richard G.**
  Poway CA 92064 (US)
- **Blevins, Terence J.**
  Springboro OH 45066 (US)
- **Ramsey, David Allen**
  Lexington SC 29072 (US)

(74) Representative: **Cleary, Fidelma et al**
International IP Department NCR Limited 206 Marylebone Road
London NW1 6LY (GB)

(54) Provision of continuous database service and scalable query performance using active redundant copies

(57) A method and apparatus for distributing computer resources in a network environment. A network of computer systems is partitioned into at least one computing system partition, and is configured into at least one redundancy group. The computing system partitions include applications, computing system nodes, and copies of a database schema. The copies of the database schema are replicated at each computing system partition within a network. The computing system partition manages interactions between the instances, the computing system nodes, and the copy of the database schema within the respective computing system partition. The redundancy group comprises at least one computing system and at a plurality of computing system partitions, and manages the replication of the database schema within the computing system and computing system partitions.

EP 0 992 909 A3

Printed by Jouve, 75001 PARIS (FR)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 30 7381

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	US 5 761 500 A (GALLANT JOHN K ET AL) 2 June 1998 (1998-06-02) * column 2, line 59 - column 6, line 37; figures 2A-2C *	1-20	G06F11/14
X	US 4 432 057 A (DANIELL THOMAS P ET AL) 14 February 1984 (1984-02-14) * column 2, line 60 - column 5, line 14; figure 1 *	1-20	
A	WO 98 26553 A (SUN MICROSYSTEMS INC) 18 June 1998 (1998-06-18) * the whole document *	1-20	
A	GARCIA-MOLINA H ET AL: "RELIABLE DISTRIBUTED DATABASE MANAGEMENT" PROCEEDINGS OF THE IEEE, IEEE. NEW YORK, US, vol. 75, no. 5, 1 May 1987 (1987-05-01), pages 601-620, XP000050672 ISSN: 0018-9219 * the whole document *	1-20	TECHNICAL FIELDS SEARCHED (Int.Cl.7)
P,A	US 5 832 225 A (HACHERL DONALD JOSEPH ET AL) 3 November 1998 (1998-11-03) * the whole document *	1-20	G06F H04L

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	26 October 2001	Bozas, I

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 99 30 7381

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

26-10-2001

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5761500 A	02-06-1998	AU 2734297 A	07-11-1997
		WO 9739413 A1	23-10-1997
US 4432057 A	14-02-1984	CA 1180458 A1	01-01-1985
		DE 3277993 D1	18-02-1988
		EP 0081056 A2	15-06-1983
		JP 1031216 B	23-06-1989
		JP 1546203 C	28-02-1990
		JP 58094046 A	04-06-1983
WO 9826553 A	18-06-1998	US 5938732 A	17-08-1999
		EP 0978184 A1	09-02-2000
		WO 9826553 A1	18-06-1998
US 5832225 A	03-11-1998	NONE	

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82